Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended) A method of processing data which is communicated over a computer network, said method comprising:

pre-allocating portions of a memory to said first processor and said second processor;

receiving first packet header data from a first network interface port,

processing ~~a first group of network packets~~ said first packet header data in a first processor which executes a first network protocol stack, and transmitting first application data associated with said first packet header data to a host processing system; ~~said first group of network packets being communicated through a first network interface port~~;

receiving second packet header data from said first network interface port,

processing ~~a second group of network packets~~ said second packet header data in a second processor which executes a second network protocol stack, and transmitting second application data associated with said second packet header data to said host processing system; ~~said second group of network packets being communicated through said first network interface port~~;

processing of a third packet header data comprising receiving third application data from a host processing system and preparing said third packet header data and causing said third application data and said third packet header data to be transmitted over said computer network through said first network interface port;

processing of a fourth packet header data comprising receiving fourth application data from said host processing system and preparing said fourth packet header data associated with said fourth application data and causing said fourth application data and said fourth packet header data to be transmitted over said computer network through said first network interface port;

synchronizing access to said memory by said first and second processors; and

maintaining a communication channel between said first processor and said second processor through a message queue.

2. (Original) A method as in claim 1 wherein said first network protocol stack and said second network protocol stack are separate processing threads.

3. (Original) A method as in claim 2 wherein said separate processing threads each comprise separate operating system software processing logic.

4. (Original) A method as in claim 1 wherein said first network protocol stack and said second network protocol stack use the same network protocols.

5. (Original) A method as in claim 4 wherein said same network protocols comprise at least one of (a) an Internet Protocol (IP) and (b) a Transmission Control Protocol (TCP).

6. (Original) A method as in claim 1 wherein said first group of network packets are associated with a first network session between a host processing system and a

first digital processing system and said second group of network packets are associated with a second network session between said host processing system and a second digital processing system.

7. (Canceled)

8. (Canceled)

9. (Original) A method as in claim 1 wherein said first network interface port comprises an Ethernet interface.

10. (Original) A method as in claim 1 wherein said first group of network packets are assigned to said first processor through a programmable hashing operation on said first group of network packets and wherein said second group of network packets are assigned to said second processor through said programmable hashing operation.

11. (currently amended) A system for processing data which is communicated over a computer network, said system comprising:

a network interface port;

a memory coupled to a bus;

a host interface port;

a first processor coupled to said network interface port, said bus and said host interface port , said first processor executing a first network protocol stack to process a first group of network packets which are communicated through said network interface port;

a second processor coupled to said network interface port, said bus and said host interface port , said second processor executing a second network protocol stack to process a second group of network packets which are communicated through said network interface port;

locking logic circuitry to synchronize access to said memory by said processors;

dispatch logic circuitry to assign a network packet to a specific one of said processors;

a DMA engine and a control queue coupled to said network interface port and said host interface port through said bus to transfer packets to and from both said network interface and said host interface;

a message queue to maintain a communication channel between said first processor and said second processor;

wherein:

said first processor to execute network protocol stack instructions to process a received first packet header data from said first network interface port; and

said second processor to execute network protocol stack instructions to process a received second packet header data from said first network interface port.

12. (Original) A system as in claim 11 wherein said first network protocol stack and said second network protocol stack are separate processing threads.

13. (Original) A system as in claim 12 wherein said separate processing threads each comprise separate operating system processing logic.

14. (Original) A system as in claim 11 wherein said first network protocol stack and said second network protocol stack use the same network protocols.

15. (Original) A system as in claim 14 wherein said same network protocols comprise at least one of (a) an Internet Protocol (IP) and (b) a Transmission Control Protocol (TCP).

16. (Original) A system as in claim 11 wherein said first group of network packets are associated with a first network session between a host processing system and a first digital processing system and said second group of network packets are associated with a second network session between said host processing system and a second digital processing system.

17. (Canceled)

18. (Canceled)

19. (Original) A system as in claim 11 wherein said first network interface port comprises an Ethernet interface.

20. (Original) A system as in claim 11 wherein said first group of network packets are assigned to said first processor through a programmable hashing operation on said first group of network packets and wherein said second group of network packets are assigned to said second processor through said programmable hashing operation.

21. (Original) A system as in claim 11 further comprising:

a first bus coupled to said first processor and to said second processor and to said network interface port;

a first memory coupled to said first bus;

a first memory controller coupled to said first bus and to said first memory, at least a portion of said first group of network packets and a portion of said second group of network packets being stored in said first memory.

22. (Original) A system as in claim 21 further comprising:

a host bus interface coupled to said first bus;

a second bus coupled to said host bus interface;

a second memory coupled to said second bus;

a second memory controller coupled to said second bus and to said second memory;

a host processor coupled to said second bus and to said second memory.

23. (Original) A system as in claim 21 wherein said first processor, said second processor, said first bus and said first memory controller are all fabricated on a single integrated circuit.

24. (Original) A system as in claim 22 wherein before said first processor executes said first network protocol stack to process said first group of network packets, said portion of said first group is stored in said first memory through a first direct memory access (DMA) operation.

25. (Original) A system as in claim 24 wherein after said first processor executes said first network protocol stack to process said first group, said portion of said first group is stored in said second memory through a second DMA operation.

26. (Original) A system as in claim 21 wherein said portion of said first group and said portion of said second group are stored in said first memory in pre-allocated portions of said first memory.

27. (Original) A system as in claim 21 further comprising:

first dispatch logic coupled to said network interface port and to said first bus, said first dispatch logic assigning said first group to said first processor through a programmable hashing operation on said first group.

28. (Original) A system as in claim 27 wherein said first dispatch logic assigns said second group to said second processor through a programmable hashing operation.

29. (Original) A system as in claim 28 further comprising:

second dispatch logic coupled to said first bus and to said host bus interface, said second dispatch logic assigning packets from said second bus to one of said first processor or said second processor.

30. (Original) A system as in claim 11 wherein said first processor and said second processor are general purpose, programmable processors.

31. (currently amended) A machine readable medium (MRM) containing executable program instructions which when executed by a processing system cause said processing system to perform a method of processing data which is communicated over a computer network, said method comprising:

pre-allocating portions of a memory to said first processor and said second processor;

receiving first packet header data from a first network interface port,

processing ~~a first group of network packets~~ said first packet header data in a first processor which executes a first network protocol stack, and transmitting first application data associated with said first packet header data to a host processing system; ~~said first group of network packets being communicated through a first network interface port~~;

receiving second packet header data from said first network interface port,

processing a ~~second group of network packets~~ said second packet header data in a second processor which executes a second network protocol stack, and transmitting second application data associated with said second packet header data to said host processing system; ~~said second group of network packets being communicated through said first network interface port~~;

processing of a third packet header data comprising receiving third application data from a host processing system and preparing said third packet header data and causing said third application data and said third packet header data to be transmitted over said computer network through said first network interface port;

processing of a fourth packet header data comprising receiving fourth application data from said host processing system and preparing said

fourth packet header data associated with said fourth application data and causing said fourth application data and said fourth packet header data to be transmitted over said computer network through said first network interface port;

synchronizing access to said memory by said first and second processors; and

maintaining a communication channel between said first processor and said second processor through a message queue.

32. (Original) A MRM as in claim 31 wherein said first network protocol stack and said second network protocol stack are separate processing threads.

33. (Original) A MRM as in claim 32 wherein said separate processing threads each comprise separate operating system software processing logic.

34. (Original) A MRM as in claim 31 wherein said first network protocol stack and said second network protocol stack use the same network protocols.

35. (Original) A MRM as in claim 34 wherein said same network protocols comprise at least one of (a) an Internet Protocol (IP) and (b) a Transmission Control Protocol (TCP).

36. (Original) A MRM as in claim 31 wherein said first group of network packets are associated with a first network session between a host processing system and a first digital processing system and said second group of network packets are

associated with a second network session between said host processing system and a second digital processing system.

37. Canceled.

38. Canceled.

39. (Original) A MRM as in claim 31 wherein said first network interface port comprises an Ethernet interface.

40. (Original) A MRM as in claim 31 wherein said first group of network packets are assigned to said first processor through a programmable hashing operation on said first group of network packets and wherein said second group of network packets are assigned to said second processor through said programmable hashing operation.

41. (Original) A method as in claim 1 wherein said first network protocol stack and said second network protocol stack use different network protocols.

42. (Original) A system as in claim 11 wherein said first network protocol stack and said second network protocol stack use different network protocols.

43. (Original) A machine readable medium as in claim 31 wherein said first network protocol stack and said second network protocol stack use different network protocols.

44. (Original) A method as in claim 1 further comprising:

processing a third group of network packets in said first processor which executes said first network protocol stack, said third group of network packets being communicated through a second network interface port.

45. (Original) A system as in claim 11 further comprising:

at least one further network interface port coupled to said first processor and said second processor.

46. (Original) A system as in claim 17 wherein said first processor and said second processor are coupled to a further host processing system.